**Tomasz Jaskiewicz** :author

TU Delft – research group Hyperbody :organization

Netherlands :country

# (In:)forming Interactive Architectural Systems, Case of the xMAiA Meta-model

:abstract

This paper positions the domain of interactive architecture (iA) and searches for an appropriate model for structure and processing of information in the design and operation of such architecture. It is shown that there are different approaches to ways in which iA system models can be defined, each with numerous advantages and disadvantages. However, due to complexity of encountered problems, application of such models can be only partially validated by simulation and hence their design is inherently dependent on creation of operational and experiential full-scale prototypes of the systems these models represent. Another observation is the lack of correspondence between existing iA models and other contemporary models of computation for architectural geometry, fabrication and engineering. A meta-model for extensible multi-agent interactive architecture (xMAiA) is consequently proposed as a remedy to this situation. xMAiA meta-model is aimed to provide an open framework for integrated evolution, development and operation of interactive architectural systems. It delivers an extensible platform, in which diverse, project-specific models and approaches can be implemented, tested, and further evolved. Such a platform has the potential to empower agile development and operation of interactive architectural ecologies, as well as to substantially facilitate integration of creative design and experiential prototyping from day-1 of project design and development cycle. An example application conforming to the xMAiA meta-model is consequently presented and illustrated with a case study project performed in the university education context.

## 1 Background

In the age of information technology, ubiquitous computing, smart materials, globalization, and the increasing need for sustainable development, architecture faces many new challenges, which propitiously come in hand with unprecedented new technological opportunities. Kas Oosterhuis, among a number of other visionary architects, has long time urged for a radically new way of approaching architecture that would allow it to embrace the new opportunities of our times:

> *Until now architecture was a discipline of intractability. Buildings were always meant to be as steady as a rock and give shape to the flow (of people, energy, and matter) and, more importantly resist that flow. Let's imagine that buildings could move with changes in use, more so than was considered possible before now, that they could move with changing conditions. (...) Buildings and built environments are becoming programmable. Form and substance can both be driven. An interactive relationship will effortlessly grow between the users and the built environment, in the way that users and smart appliances are beginning to communicate now. Buildings will develop into a smart swarm of building parts in contact with each other and with their users"* (Oosterhuis, 2001).

What was a daring vision a decade ago, now slowly becomes reality. Clearly, architecture is already experiencing a shift of paradigm *(Kuhn, 1996)*, or in Marcos Novak's words, architecture "transverges" *(Novak, 2006)* into a new kind of profession, being driven by the information technology revolution *(Saggio, 2007)* and latest developments in science, humanities, and arts.

iA's roots may be traced back to modernist concepts of buildings being "machines for living in" (Le Corbusier) and "open plan" (Frank Lloyd Wright) *(Conrads, 1975)*. Nevertheless, true explosion of visions for dynamic architectural structures took place in late 1950s and 1960s. Projects such as Yona Friedman's "Spatial city," Constant's "New Babylon," Archigram's "Plug-in city," "Seaside bubbles," or "living pods," or imaginative city systems of the Japanese metabolists, all present a vision of architecture that constantly evolves in relation to the needs of its inhabitants. Eventually, Cedric Price's "Fun Palace" *(Price et al., 2003)* became the seminal case illustrating such vision in a technologically realizable proposal on an architectural scale. Price's project was advised by Gordon Pask. Pask's works in cybernetics,

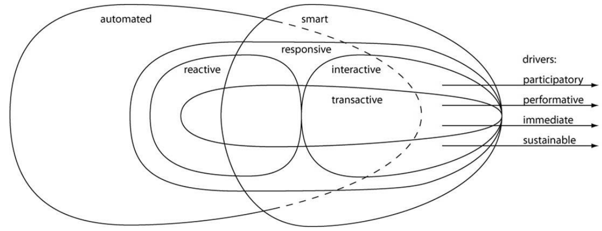


Figure 1. Synthesis of domains and drivers of dynamic architecture.

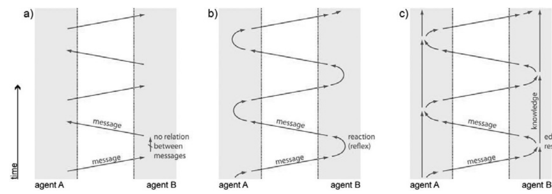


Figure 2. Information flow (red arrows) in a) two-way, b) reactive, and c) interactive communication,

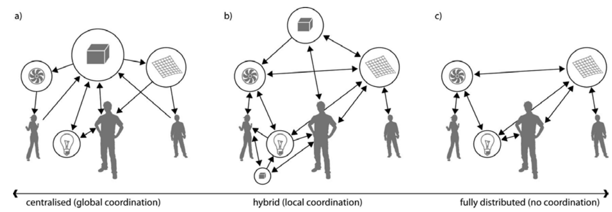


Figure 3. Information flow in centralized (a), hybrid (b), and distributed (c) concepts of user interactions with architecture (box icon represents the system coordinator/controller; other icons represent active sensing and/or effecting building components).

especially his conversation theory and his understanding of architecture as a system *(Pask, 1969)*, became highly influential for much of the following work in the domain of responsive architecture. Among many projects influenced and advised by Pask are experiments of Nicholas Negroponte *(Negroponte, 1973)* carried on in 1960s and 1970s, and later of John Frazer *(Frazer, 1995)* in 1980s and early 1990s. In theirs and other related works, the concept of modernist open plan is replaced by "non-plan" *(Hughes & Sadler, 2000)*, a blueprint of an architectural system not predetermining its own future evolution. Currently, many creative explorations in the domain of responsive architecture continue to be performed on the border of architecture, art, and engineering, typically in the form of responsive installations. Projects of Usman Haque, Philip Beesly, Tristan d'Estree Sterk or Kas Oosterhuis' Hyperbody group are some among many examples.

## 2 Interactive Architecture (iA)

Over the last decades, numerous and often fairly synonymous terms have been used to label visions of dynamically changing architecture. Many of these notions, such as "smart architecture" *(Senagala, 2006)* or "responsive architecture" *(Beesley et al., 2006)* denote the kinds of systemic mechanisms by which concerned dynamic architecture is envisioned to operate. Others, such as "performative architecture" *(Kolarevic, 2005)* or "immediate architecture" *(Friedrich, 2008)*, refer to specific qualities of architecture that justify the need for existence of virtually or physically dynamic architectural systems.

To position the notion of interactive architecture among these terms, the concept of interactivity first needs to be clarified. Following S. Rafaeli *(Rafaeli, 1988)*, there is a clear distinction between reactive communication (sent message is only affected by one, immediately preceding it, received message) and interactive communication (messages not only sequentially react to each other, but also depend on the history of earlier received messages). The term responsive covers both kinds of communication processes.

In this context, interactive architecture (iA) denotes a subset of responsive architecture. While responsive architecture can also mean architecture that just reacts to users' actions, interactive architecture strictly implies a reciprocal relationship between architecture and its inhabitants and an inherent learning process of both. In consequence of this understanding of interactivity and in synchrony with other definitions (e.g. of Haque *(Haque, 2006)*, Oosterhuis *(Oosterhuist & Xia, 2007)*, Fox, and Kemp *(Fox & Kemp,2009))*, iA can be defined as architecture that exhibits autonomous behavior, in which that behavior evolves through interactions with its users and environment.

## 3 Models for iA

One of the key questions to be answered when conceptualizing interactive architecture is how information flow is to be organized in iA? One aspect is the flow of information within the interactive architecture itself. The other, more significant, aspect is the flow of information between iA and its users. Culturally, we are accustomed to the idea of humans controlling all artificial systems. The notion of control, however, refers to a one-directional mode of communication, which stands in opposition to the idea of interactivity that entails an inherently reciprocal relationship. Nevertheless, the following models of control in architecture are by analogy also applicable to interactive architecture, assuming that the relationship of control can be substituted by the relationship of mutual affect.

Two elementary models for user control of participatory architecture (both presented at the same conference in 1971) belong to Yona Friedman, the "Participatory Architecture" *(Friedman, 1972)* model, and Charles Eastman, the "Adaptive-Conditional Architecture" *(Eastman, 1972)* model. Friedman's model puts building users in direct control of architectural adaptations, through one, centralized interface and control system, which coordinates all architectural adaptations within the given context. Conversely, Eastman's model proposes a distributed system of installations and devices that operate autonomously and are individually controlled by building users through local (negative) feedback loops. Tristan d'Estree Sterk reintroduced models of Eastman and Friedman to the contemporary discourse and proposed another model combining the two, which he calls the "hybridized model of control" (d'Estree Sterk, 2006), where both local and global (top-down and bottom-up) control is possible (enforced either by users or by "intelligent processes"). By generalizing this classification, interactive system models can be ordered along the line of their increasing distribution and flexibility and decreasing centralization (shift from global, through local, to no coordination).

Any kind of such models requires communication between at least some of its autonomous components, and thus can be treated as a multi-agent system (MAS). MAS are typically developed in software, but can also be found in robotic applications (e.g. Swarm-bots *(Mondada et al., 2004)*) and are often associated with development of artificial intelligence *(Russell & Norvig, 2009)* (e.g. swarm intelligence). MAS can consist of simple-reflex or learning (intelligent) agents. Many of the existing MAS models include coordination or control mechanisms operating in parallel to distributed agent processes. There are numerous established models of such "hybridization," with varying degrees of importance of coordination or control, listing of which is beyond the scope of this paper. Generic MAS models are thus an obvious foundation for development of more specific models for iA.

However, recent decade has also seen substantial developments in building automation models *(Wang, 2009)*, building information models (BIM) *(Eastman et al., 2008)*, computational models for complex architectural form generation *(Aish & Woodbury, 2005)*, engineering and digital fabricating *(Iwamoto, 2009)*. These models also need to be addressed and possibly integrated in the development of iA.

Overall, it can be stated, that although there are many possible models that could be potentially applied to creation of iA or its subsystems, little work has been done to systematically evaluate, test, and integrate these models in context of their validity for iA. In most cases, integration of such models requires custom-engineered hardware and software, which would allow dynamic changes of inherent architectural qualities (including building form), deliver necessary heuristics, while also guaranteeing safe and reliable operation. Consequently, the cost and complexity of creating full-scale interactive buildings still appears prohibitive and hinders commercial application of iA. There is also not enough measurable evidence for the explicit added value and performance of potential interactive "whole-building" systems to provide substantial interest for them from the commercial sector. It is clear that in order to set further development of iA in motion, considerable amount of experimentation is needed to creatively formulate and test various models and their performance.

## 4 xMAiA: extensible multi-agent interactive architecture meta-model

Based on empirically evaluated projects and case studies *(Jaskiewicz, 2008)*, it has been concluded that there is need for generic system architecture that could support the rapid creation of operational and experiential prototypes of interactive architectural systems from the initial moment of their design processes. What's more, not only implementation, but also the architecture of such systems needs to be able to evolve over time, in-line with development of project-specific technological solutions. Expanding on those ideas is the following requirements for an iA model.

*The model:*

*1) Provides a representation of a multi-agent, complex adaptive system constituting a complete architectural environment within a defined boundary. (Since any kind of entity can be abstracted as an agent, a boundary needs to be drawn, otherwise the system extends to infinity.)*

*2) Is abstract and independent of used technology. (Technology-specific limitations should not determine the architecture of the system, although they are bound to determine its implementation. Implementation may however change over time.)*

*3) Is easy to comprehend for non-proficient users, while providing exhaustive and fully operational descriptions of its components. (Not only designers, but also daily users need to be able to easily comprehend iA systems in which they themselves need to operate and interact with.)*

*4) Supports creation of embodied virtual and physical agents, as well as abstract agents. (Not only physical agents can exist in iA systems. Virtual agents are necessary for definition of much of non-embodied spatial qualities and abstract agents are necessary to define parts of the system which do not have an explicit location in space.)*

*5) Consists of agents that are individually identifiable, that are described by properties, that hold autonomous behaviors, and that can be mutually related and communicate with each other. (These qualities are necessary for a minimal, working definition of an agent.)*

*6) Consists of agents that can transform and evolve over time. (In order to allow simultaneous development and evolution of an iA system.)*

*7) Consists of agents that can be nested; one agent can represent a subsystem of multiple other agents. (Depending on the given context, an assembly of agents can be treated as one agent; in an oversimplified example, a house can be treated as an assembly of roof and walls, a wall can be seen as an assembly of individual bricks, etc.)*

*8) Delivers a location model (McCullough 2005) for its constituent agents and allows development of hybrid distributed and coordinated systems. (Location models provide a context for agents to operate within, where the context can have a coordinating role in otherwise distributed systems. A location model can thus be provided by specialized kinds of coordinating agents.)*

*9) Does not enforce any particular classification of agents, but permits development of such classification. (In the longer perspective, a fixed classification of agents is likely to hinder evolution and development of an iA system over time. On the other hand, ad hoc classifications are often necessary in order to facilitate efficient development and system implementation; not every agent can be designed and created individually.)*

*10) Is modular. (Ontologically, the model allows for different fragments of an iA system to be disconnected and assembled in different configurations or embedded in other iA systems. One system may also partly overlap with another, be contained or contain another system.)*

*11) Supports creation of physical and virtual assemblies of agents and their integration. (Virtual (localized and abstract) and physical agents need to operate in synchrony.)*

*12) Supports representations to evolve into becoming the actual systems they represent. (There can be no explicit separation between design and operation of an iA system. Thus, the division between the representation and what it represents is bound to disappear. A model needs to become an integral part of what it represents.)*

It can be easily concluded that the following requirements cannot be easily satisfied by one model, but several parallel models are likely to be required. What's more, these models are likely to change over time and may radically differ between specific projects. An integration of such diverse models can be achieved by a meta-model (or "model of models"). The purpose of such a meta-model

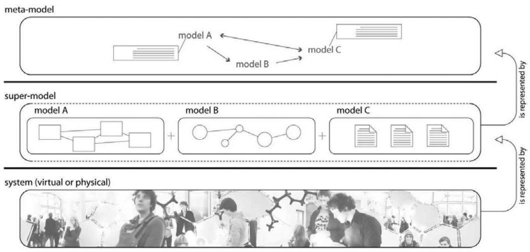


Figure 4. Concept of a meta-model for iA.

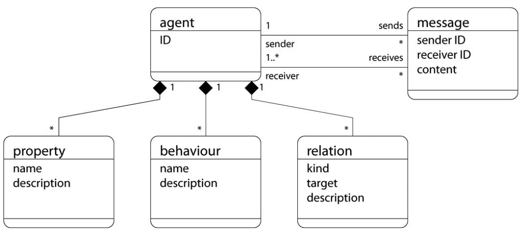


Figure 5. Layer 0 class diagram (UML).

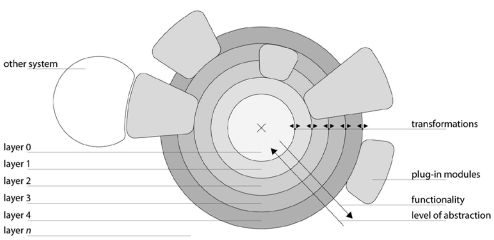


Figure 6. xMAiA model, layered structure.

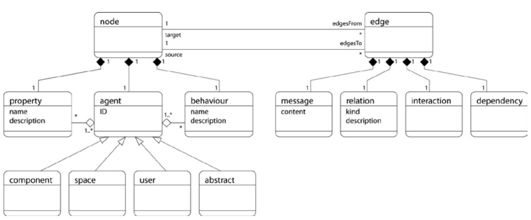


Figure 7. An example of more specific class structure implemented in layer 1 (UML). Its content is bi-directionally transformable to layer 0 ontology, although it is not an extension of layer 0 class structure.

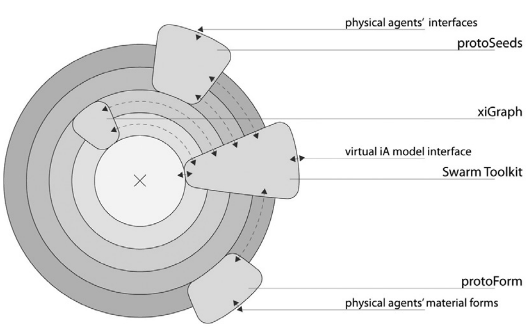


Figure 8. Implementation of preliminary set of plug-ins in the xMAiA meta model (arrows denote flow of information in the meta-model).

is not to compete with any of the specific models. On the contrary, its objective is to facilitate integration and further development of a variety of specialized models by providing an easily comprehensive, non-constraining framework for binding these models, while being inherently independent of these models and of any other standards. Its main purpose is thus, to empower creative and out-of-the-box experimentation with design of interactive architectural environments and their immediate deployment in the physical domain, with an option of flexible, project-specific, plug-in connectivity to any of the required technological solutions and standardized systems (if explicitly needed). In that respect, rather than providing a means for finding technological answers to well-defined problems, the developed meta-model should be aimed at establishing a platform for exploration of yet undefined problems inherent to the still vacillating path of further development of interactive architecture.

The proposed answer to those demands is a layered meta-model of extensible multi-agent interactive architecture (xMAiA). Each layer of xMAiA represents a comprehensive iA model. A model containing multiple layers can be referred to as a super-model. On the other hand, a layer model can contain sub-models. Within this structure, the core layer 0 is the only permanently defined, minimal model, which allows an abstract, but also complete description of all agents featured in a system. Each agent has a unique identification; it can contain properties (parameters), behaviours, and relations to other agents. It can also send and receive messages to and from other agents. There are no elements in the system other than agents and their components.

The number of other layers can vary. The higher the layer's number, the less abstract and more functional it is expected to be and more frequently it is expected to change. All information contained in the model has to be translatable between adjacent layers in both directions. The following guidelines are the initial distribution of problems to be addressed in consecutive layer models in their first implementation template.

- Layer 0: model describing main features of agents in the system (permanently defined)
- Layer 1: model describing operational ontology of individual agents in the system (may vary across projects)
- Layer 2: model describing types and operation of multi-agent assemblies, e.g. hierarchies, coordination, communication, dependencies (is likely to change multiple times within one project)

- Layer 3: model describing operation of agents over hardware layer, as well as compliance with chosen protocols and eventual standards (defined when project reaches required complexity, can be frequently updated and modified)
- Layer 4: operational model – defines ways to describe all physical elements needed for the model to operate in real world, assembly and fabrication of their components (needs to be adjusted for every kind of material components, electronics, sensors, actuators, production processes used, etc.)
- Layer n: the last layer of the project is the actual physical environment in which the project is designed to operate.

A general guideline for layer 1 is thus to include further classification of agents. Proposed and initially predetermined taxonomy includes material components, spaces (locators), and users, as well as abstract agents, which may represent, e.g., assemblies of agents of diverse kinds, local coordinators, legal entities, etc. Notably, human users are treated here as agents inherent in the system. The following layer models are defined more flexibly.

Specific functionality operating across multiple layers can be encapsulated in independent plug-ins. Such plug-ins provide abstractions for complex functionalities and can be reused between applications (thus also between different super-models and templates). They may include hardware and software elements and can handle transformations of information between and within layers.

## 5 xMAiA – first template

The first template of a complete and functional set of xMAiA layer models and plug-ins is minimal, but sufficient for the system's complete operation on all design and prototyping stages. Layer 1 (defined as in figure 7) is extended with the plug-in software design environment Swarm Toolkit (Jaskiewicz, 2008), which allows quick creation and instantiation of virtual agents of any sub-class. This plug-in is also used to handle all translations between layers. Data model XiGraph *(Oosterhuis et al., 2007)* is used as an underlying data structure composed of nodes and semantic edges. 3D VIA Vitrools (3D VIA Virtools is a software development and deployment platform; http://www.3dvia.com/products/3dvia-virtools) scripting language (VSL) is used to describe agent behaviours. On layer 2, a simple message format is specified to communicate between agents. The protoForm plug-in is provided to facilitate rapid CNC fabrication of selected parts of physical agents

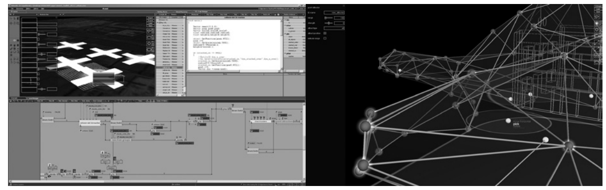

Figure 9. Swarm Toolkit software implemented in 3D VIA Virtools platform allows to easily model virtual agents and test multi-agent systems in virtual space.

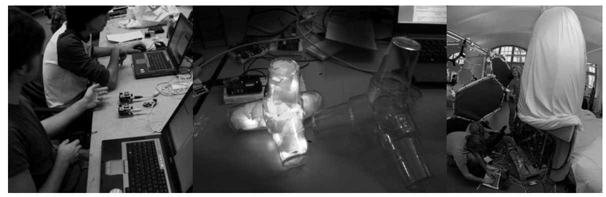

Figure 10. Layer 3, early prototypes of embedded agents are synchronized with the virtual model.

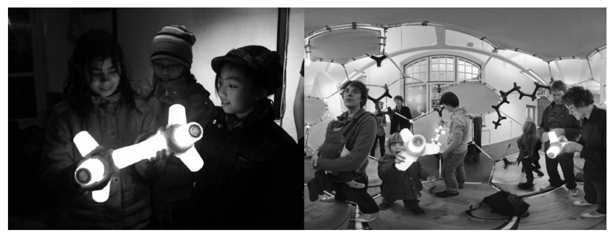

Figure 11. Layer 4; embedded agents are operating autonomously (sCAPE project).

Figure 12. Kinetic actuation of structure as further evolution of the working installation (Odyssey project).

by directly using selected properties of individual agents to control parametric solid models and automatically derive fabrication data that can be sent to a CNC machine. Layer 3 introduces Adruino-based (Arduino is an open-source physical electronics prototyping platform; http://www.arduino.cc) embedded agents called protoSeeds as another plug-in. Each protoSeed can be directly paired to a virtual agent created in the Swarm Toolkit software and augment it with physical sensing and actuation. Layer 4 extends this model with introducing a transformation of complete agent behaviour into the code of an embedded agent (translation from VSL to Arduino code), thus removing that agent from the virtual model.

## 6 Application

The examples presented further are based on the work developed by an interdisciplinary group of students in the course of the Interactive Environments Minor at Delft University of Technology. The course was a cooperative effort of research group Hyperbody at the faculty of Architecture, and ID-StudioLab at the faculty of Industrial Design Engineering. (Students in the course were: Teun Verkerk, Tom Goijer, Cees-Willem Hofstede, Iris van Loon, Marieke Dijksma, Thomas Oekelen, Fons van der Berg (sCAPE group), Merijn Pen, Lieke Kraan, Govert Flint, Tomas Verhallen, Bob Groeneveld, Melisa Garza-Vales, Jesse Timmermans (Odyssey group), Altynay Imanbekova, Joris Hoogeboom, Ben van Wijk, Hanneke Hoogewerf, Tino de Bruijn, Patrick Pijnappel, Alexander de Mulder (GEN group). Leading teachers were: Walter Aprile, MarkDavid Hosale, Aadjan van der Helm, Dieter Vandoren and the author.) Their work has been adapted to serve as a set of examples for the application of xMAiA.

A team of designers and engineers is given a theme and context of requested architectural intervention (themed as a lounge environment located in a public setting). While they brainstorm about their first ideas on how an interactive lounge should behave, they use the given template and define the first types of autonomous agents of which the system envisioned by them should comprise. The team instantly creates virtual models of these agents and formulates their initial, simple behaviors, relations, and mutual dependencies. All of these activities are performed using the Swarm Toolkit platform.

While the brainstorming continues, more elaborate behaviors and multi-agent assemblies are created and virtually validated. Simultaneously, first physical parts of agents are rapidly fabricated from transparent Perspex, using a laser cutter with help of the protoForm plug-in. Created physical components become augmented with simple sensors, actuators, and protoSeed units. Once paired with selected virtual agents, physical objects begin to augment the behavior of the corresponding virtual agents in the physical world, in real-time.

This process continues iteratively. The prototyped installation becomes more complex, additional constrains and features are added. Some agents are removed, many others are introduced. More material elements are also added to the installation. Users are represented in the virtual model by avatars and can remotely interact with virtual model components through these avatars. Testers can also begin to experience some of the working parts of the system directly, in the physical world.

Gradually, the behavior of selected agents becomes translated into the code of protoSeeds. Such embedded agents communicate now with other agents only where a physical communication layer permits them to. Their representation in the virtual model indicates the change of their state. The hierarchy inverts; now the corresponding virtual agents are controlled by the physical ones. The entire installation becomes fully operational. Eventually, it can be tested by granting access to the wider public. However, the design process doesn not stop there. In fact, it never has to end. The installation can be continuously reconfigured and changed. Other types of actuation and sensing can be added. Ultimately, the installation evolves to become a full featured building.

## 7 Conclusion

Clearly, the proposed xMAiA meta-model is, to a certain degree, idealistic and contains several risks and threats. Nevertheless, it does provide an answer to all initially set challenges—among them to facilitate the development of contextualized-in-space, complex interactive environments, as well as to meet the need for an extensible framework for integrated design and prototyping of such dynamic architectural systems.

It is acknowledged that the lack of standards and no predefined structure for individual layer models may (and arguably, should) lead to large inconsistencies among different future implementations developed under the xMAiA meta-model framework. However, to date, most of the experimental projects in the domain of iA do not conform to any common models. Thus, providing a meta-framework without any substantial constraint to freedom of creative exploration seems to be the only solution to allow further development of experimental models, while also providing a platform for interoperability and reuse of components, sub-models, and plug-ins across projects when desired. To further facilitate such an exchange, an on-line service is planned for introduction, to serve as a hub for the exchange of super-model templates, plug-ins, and showcase of projects. In this way, xMAiA has the potential to provide an open platform, which can support emergence of common models, but does not enforce any standards and encourages evolution and variation among and throughout all of its implementations.

On the technical side, in the implementation of a specific xMAiA super-model, translations between layers can be problematic and may need to be solved individually in each project. This allows more freedom in system formulation, but also requires an overhead of work in the implementation phase. This is considered to be a necessary trade-off for the xMAiA model's flexibility. However, along with new applications, further specification of conventions for inter-layer translations is bound to occur. Encapsulation of translations in open-source plug-ins is an intermediate solution that retains flexibility, while automating and hiding transformations between layer models.

Further development of the xMAiA may only happen through its application to projects. This would ultimately lead to the emergence of numerous new operational templates (common models) and further refinement of the xMAiA meta-model and its implementation guidelines. In the more distant future, an open community is hoped to emerge around the application of xMAiA, driving further the creative development of iA and the following creation of rich and dynamic spatial ecologies, providing unprecedented spatial qualities and experiences.

**References**

Aish, R., & Woodbury, R. *(2005). Multi-level Interaction in Parametric Design. In Smart Graphics. Lecture Notes in Computer Science. Heidelberg: Springer Berlin.*

Beesley, P., Hirosue, S., & Ruxton, J., *(2006). Responsive Architectures: Subtle Technologies, Riverside Architectural Press.*

Conrads, U., *(1975). Programs and Manifestoes on 20th-Century Architecture, The MIT Press.*

d'Estree Sterk, T. *(2006). Responsive Architecture: User-Centered Interactions. In Game Set and Match II: The Architecture Co-laboratory on Computer Games, Advanced Geometries, and Digital Technologies, eds. Oosterhuis, K., and Feireiss, L., Rotterdam: Episode Publishers.*

Eastman, C. *(1972). Adaptive-conditional Architecture. In Design Participation, Proceedings of the design Research Society's Conference. Manchester: Academy Editions.*

Eastman, C., Teicholz, P., Sacks, R., & Liston, K., *(2008). BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors, Wiley.*

Fox, M., & Kemp, M., *(2009). Interactive Architecture, Princeton Architectural Press.*

Frazer, J. *(1995). An Evolutionary Architecture, Architectural Association Publications.*

Friedman, Y. *(1972). Information Processes for Participatory Design. In Design Participation. Proceedings of the Design Research Society's Conference. Manchester: Academy Editions.*

Friedrich, C. *(2008). Information-Matter Hybrids - Prototypes engaging immediacy as architectural quality. In What Matter(s)? First International Conference on Critical Digital. Cambridge: Harvard Graduate School of Design.*

Haque, U. *(2006). Architecture, interaction, systems. In Arquitetura & Urbanismo, vol. 149, Brazil.*

Hughes, J. & Sadler, S. *(2000). Non-Plan: Essays on Freedom, Participation and Change in Modern Architecture and Urbanism, Architectural Press.*

Iwamoto, L. *(2009). Digital Fabrications: Architectural and Material Techniques 144th ed., Princeton Architectural Press.*

Jaskiewicz, T. *(2008). 'iPortals' as a Case Study Pre-Prototype of an Evolving Network of Interactive Spatial Components. In Proceedings of the 28th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA), Minneapolis.*

Kolarevic, B. *(2005). Performative Architecture: Beyond Instrumentality, Routledge.*

Kuhn, T.S. *(1996). The Structure of Scientific Revolutions, University Of Chicago Press.*

McCullough, M., *(2005). Digital Ground: Architecture, Pervasive Computing, and Environmental Knowing, The MIT Press.*

Mondada, F. Pettinaro, G., Guignard, A., Kwee, I., Floreano, D., Deneubourg, J., Nolfi, S., Gambardella, L., and Dorigo, M., *(2004). Swarm-Bot: A New Distributed Robotic Concept. In Autonomous Robots, 17(2):193-221, Kluwer Academic Publishers.*

Negroponte, N. *(1973). The Architecture Machine: Toward a More Human Environment, The MIT Press.*

Novak, M. *(2006). Transvergence: Finite and Infinite Minds. In Game Set and Match II: The Architecture Co-laboratory on Computer Games, Advanced Geometries, and Digital Technologies, eds. Oosterhuis, K., and Feireiss, L., Rotterdam: Episode Publishers.*

Oosterhuis, K. *(2001). Architecture Goes Wild: Manifest Writings, 010 Uitgeverij.*

Oosterhuis, K., Friedrich K., Jaskiewicz T., & Vandoren D. *(2007). Protospace Software, In Second International Conference World of Construction Project Management. Delft, the Netherlands: TU Delft.*

Oosterhuist, K., & Xia, X., *(2007). iA #1, episode publishers.*

Pask, G. *(1969). The Architectural Relevance of Cybernetics. In Architectural Design.*

Price, C., Keiller, P., & Obrist, H.U. *(2003). Re: CP., Birkhäuser Basel.*